CS2310 Multimedia Software Engineering

K1 Project Report

Zihao Zhao

December 15, 2014

**Introduction**

In this class project, a home healthcare system with fall detection and speed warning is devised. It is implemented on the Slow Intelligence System(SIS) with Microsoft Kinect to keep track of patient movement and send warning speed messages to online health care center to update the status of patient.

Slow Intelligence Systems is a system that follows on the idea of slow intelligence which is the ability to step back and calculate possibilities without any outside constraints. It solves problems by trying different solutions many times, adapt to different situations and propagate knowledge. It may not perform well in the first place com- pared to fast intelligence system, but in the long term, its performance improve greatly.

The Microsoft Kinect is a device that generates a depth and image of a human body. It can be used as input sensor of SIS health care system to have a vision of patient movement. In the SDK of Kinect, Microsoft has provided us a tool to keep track of a skeleton of human body and abstract it into 3 dimensions coordinates. By these coordinates, we can easily develop algorithm to perform speed estimation and fall detection.

**System Overview**

The system can be described as 6 main components:

**KinectInput** :  
The KinectInput component is responsible for the data input. It takes Kinect Skeleton data stream and convert it to coordinates of 12 parts of human body. It generates MSG 43 and send it to Kinect component via SIS Server.

**Kinect** :  
The Kinect component mainly implements the fall detection and speed estimation algorithm. On receiving MSG 43 from KinectInput component, it runs the core algorithm to judge whether there is a fall or high speed movement of patient, if it does, it generates MSG 44 to SIS Server, and the server forwards it to Uploader component to update the status of online database.

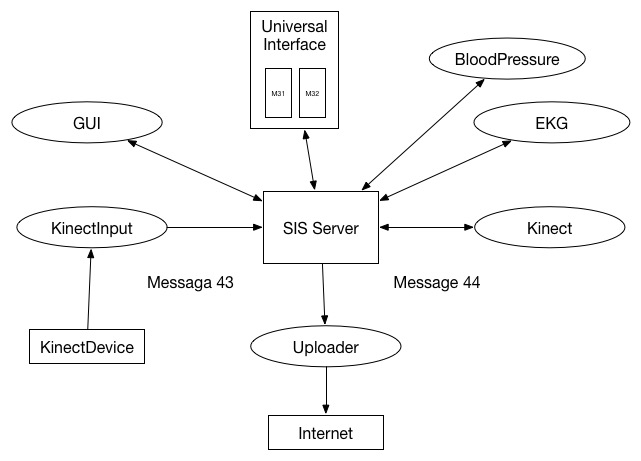
**Uploader** :  
The Uploader component receives MSG from SIS server and sent messages to online database to change the status of the patient. In our system, Uploader gets MSG 44 Fall Detection and Speed Warning messages from Kinect component and send Email to doctor, where the patient can get help.

Figure 1: SIS System

**Core Algorithm**

**Person Detection**

To detect and track a person, the skeleton stream from the Kinect is used. An event is registered to listen and track the skeleton frame. Once a person is detected, the proposed algorithm starts processing the skeletal information.

**Human Body Skeleton Representation**

The Kinect for Windows SDK provides a skeleton-tracking feature that allows developers to recognize people and track their actions. Using depth sensors, the Kinect can recognize up to six users who are standing between 0.8 to 4.0 meters away. Two of the detected users can be tracked in detail with twenty joint positions. Each skeleton joint is measured in a three dimensional (X, Y, Z) plane.

**Fall Detection**

The proposed method to detect a fall is based on the skeleton joint positions relative to the ground. Since the proposed fall detection algorithm calculates the distance between the body joint points and the floor-plane, the floor-plane must be detected first. The Kinect for Windows SDK provides a floor-clipping-plane vector, which contains the

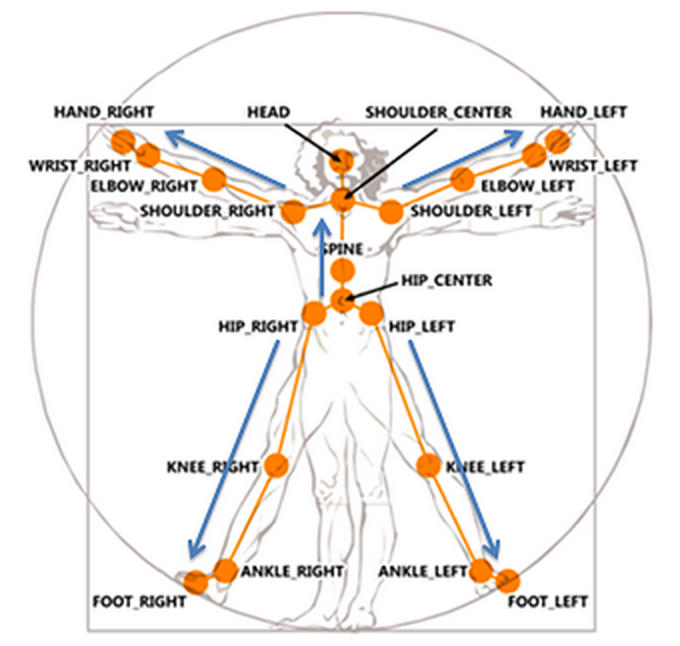


Figure 2: Skeleton Joint Points

coefficients of an estimated floor-plane equation. To calculate the distance between a point and the plane, when the point is (x0, y0, z0) and the equation of the plane is (*ax* + *by* + *cz* + *d*), Eq.(1)is below, where *D* is the distance between the point and the plane.

Using this relation, the distance from the floor to each body joint point can be calculated. A fall is detected by thresholding the distance between the joint points that are shown in the header of MSG 43. (head, hand\_left, hand\_right, elbow\_left, elbow\_right, shoulder\_left, shoulder\_right, hip\_center, knee\_left, knee\_right, foot\_left, foot\_right)

The fall detection algorithm considers the distances only for joints which are tracked by the Kinect. If every tracked joint has a normal distance less than some threshold the algorithm sets the state to fall, otherwise not fall. The algorithm 1 show the basic idea.

**Speed Estimation**

The Kinect provides approximately 30 frames per second of data. From each frame we use the timestamp (in milliseconds) and the 3D coordinate location of each joint. We also use the angle of the Kinect sensor which is assumed to not change throughout any calculations. Our algorithm also assumes that the Kinect is placed on a level surface.

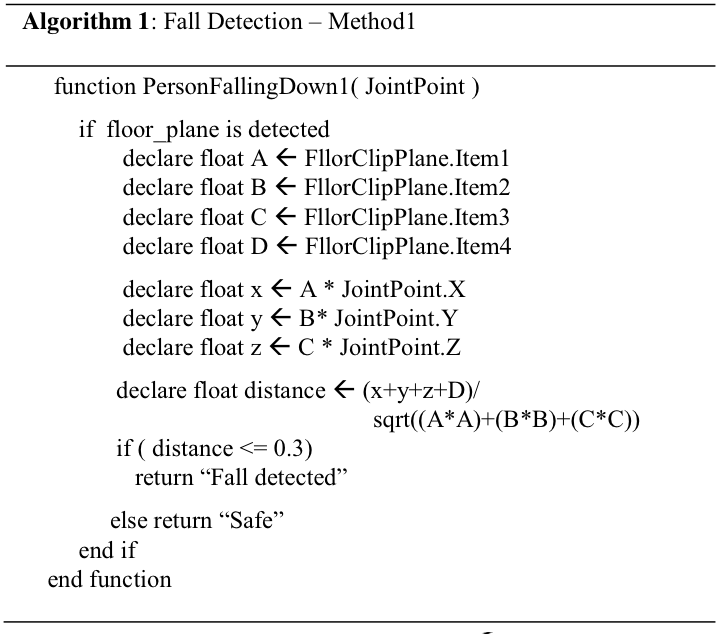
Instead of using the floor plane equation provided by the Kinect (this is not always detected, particularly on stairs) we calculate our own floor plane. If we assume the Kinect is on a level surface then we can calculate the floor plane equation from the

Figure 3: Fall Detect Algorithm

angle of the sensor as follows:

where

*Ax*+*By*+*Cz*+*D* = 0

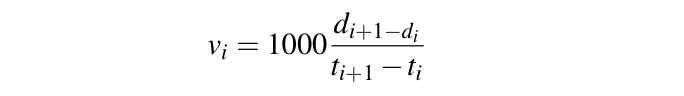
*A*=0

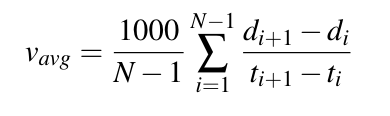
*B* = cos θ

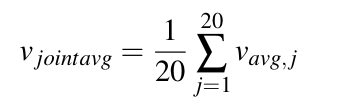
*C* = sin θ

*D*=3

A, B, and C are simply the vector normal to the floor and D shifts the floor plane 3 meters below the Kinect. The distance from the floor plane can then be calculated using Eq.1.

For frame *i* and *i* + 1 the speed for a particular joint normal to the floor is then:

Where *t* is the timestamp in milliseconds. The factor of 1000 allows us to work in more convenient units of meters per second instead of meters per millisecond. This velocity is averaged over *N* frames:

If a joint is not tracked for frame i or i + 1, the velocity is not used and the value of N for the joint :

If the speed is higher than the threshold, the system will detect a fast movement of

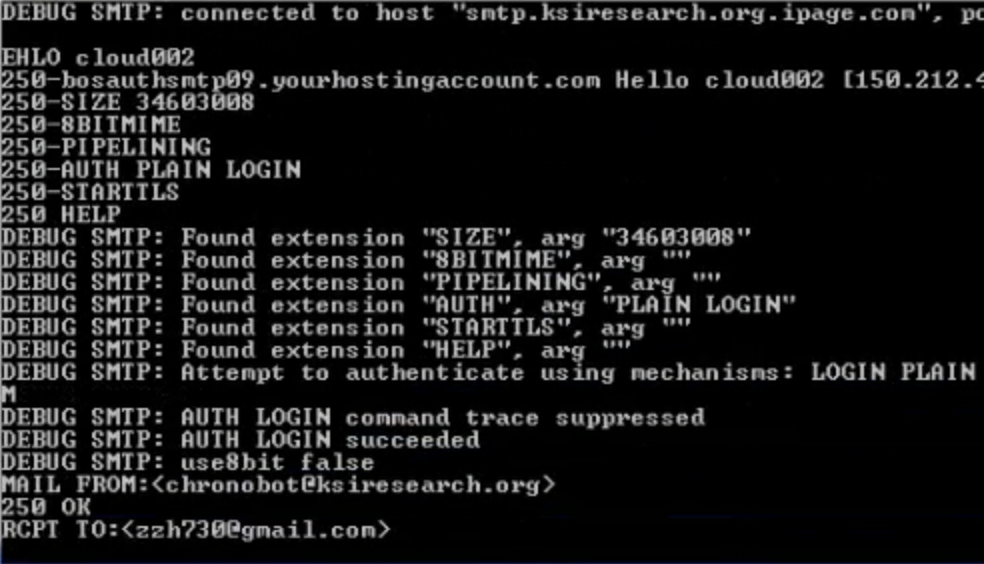
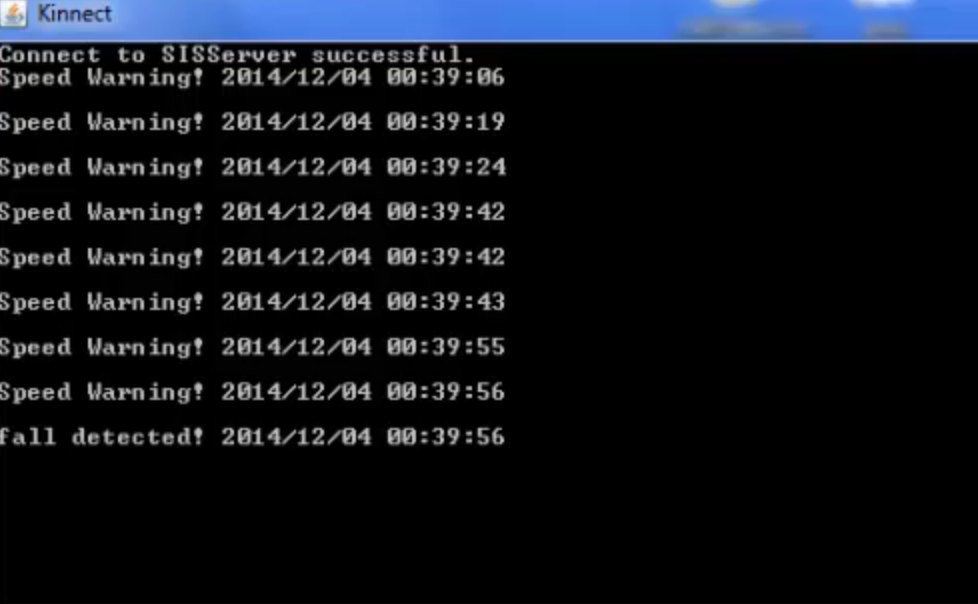
patient and sent the Kinect warning message.

**Implementation**

The whole project is based on the SIS health care system. The Kinect sensor is use to implement on the computer contains the SIS system to monitor the patient movement. We just launch the SIS testbed and use Kinect sensor as input, and if it receives any warning message, we forward it to Gmail as alarm to the doctor.

**Results and Conclusion**

Experiments were conducted in a real indoor environment under same light condition. First the system successfully recognize human body movement and abstract them into 3D coordinates. Then it can detect patient fall and fast dangerous movement. Finally, after the detection, it will sent message to via Email. So the basic frame is achieved successfully. The result can be seen from figures and there is a demo video on YouTube.



**References**

. [1]  S. Abbate, M. Avvenuti, P. Corsini, A. Vecchio, and J. Light, ?Monitoring of hu- man movements for fall detection and activities recognition in elderly care using wireless sensor network: a survey,? in Wireless sensor networks: application- centric design, Y. K. Tan, Ed. China: InTech, 2010, pp. 1-20.

. [2]  A. Mihailidis, B. Carmichael, and J. Boger, ?The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home,? IEEE Transactions on Information Technology in Biomedicine, vol. 8, issue 3, pp. 238?247, Sept. 2004.

. [3]  Y. Huang, S. Miaou, and T. Liao, ?A human fall detection system using an omni- directional camera in practical environments for health care applications,? pre- sented at the MVA2009 IAPR Conference on Machine Vision Applications, Yoko- hama, Japan, May 20-22, 2009.

. [4]  R. Planinc and M. Kampel, ?Detecting falls by using depth cameras,? presented at the 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012.

. [5]  C.Rougier,E.Auvinet,J.Rousseau,M.Mignotte,andJ.Meunier,?Falldetection from depth map video sequences,? in Toward Useful Services for Elderly and People with Disabilities, vol. 6719, B. Abdulrazak et al. Eds. Montreal, Canada: Springer Berlin Heidelberg, 2011, pp. 121-128.

. [6]  M. J. Khan and H. A. Habib. Video Analytic for Fall Detection from Shape Fea- tures and Motion Gradients. Proceedings of the World Congress on Engineering and Computer Science 2009 Vol II WCECS 2009, October 20-22, 2009, San Fran- cisco, USA.